0 EDUPACK LiU.CVL.ORIENTATION2

Version: October 29, 2003

By Michael Felsberg.

0.1 Summary

The material presented in this package is focused on representation and estimation of local orientation in multi-dimensional signals using other methods than the at CVL developed tensor approach. Hence, this package should be considered as an add-on to [6].

By now, the package just contains a theory part. An exercise part and a computer exercise part will be added in the near future.

0.2 Scientific Background

The material in this package is a compilation and adaptation of various publications, books, and reports for use in our computer vision undergraduate courses. The foundation of this material are the following publications

- Bernd Jähne's lecture notes in his course book (English edition) [5].
- The habilitation thesis of Wolfgang Förstner (in German) [3].
- Gerald Sommer's lecture notes (in German, unpublished).
- The text books of Athanasios Papoulis [7, 8, 9].
- The PhD thesis of Michael Felsberg [2] and the therein cited references.

0.3 Acknowledgement

Special thanks to Klas Nordberg for support concerning the EduPack system. Please send feedback to the authors if you have comments or find errors.

0.4 Prerequisites

In order to comprehend this package you will have to be familiar with the following concepts:

- From linear algebra: vector spaces, scalar products, bases and dual bases, vectors and matrices, decomposition of symmetric matrices using eigenvalues and eigenvectors.
- From signal processing: the Fourier transforms of multidimensional functions. In particular the concepts of simple signals and the properties of their Fourier transforms. See chapter 4 of [4].
- Edupack: Orientation [6] should be read thoroughly.
- For some parts of this package it might be useful to have some basic knowledge about stochastic signal processing. Basic knowledge about probability theory (stochastic variables, probability density functions, expactation value, moments) is required.

0.5 Feedback

This package is a dynamic document, and we invite you to send feedback to the authors in order to make it dynamic. If you find errors, or have a question, or think that some part of the text is unclear, it is likely that your comments will modify the package in some way or another.

Send comments to

mailto:mfe@isy.liu.se

1 Introduction

This package Edupack: Orientation2 should be considered as an add-on package to [6]. The main issue of this package is to introduce alternative approaches to those described in [6]. Some additions about basic issues (intrinsic dimensionality, stochastic processes) are made in the sections 1.1 and 1.2 of this introduction chapter.

Other, more involved additions are made in separate chapters:

- The structure tensor based on outer products of gradients: chapter 2.
- Isotropic quadrature filters: chapter 3.
- Multiple orientation representation by channels: chapter 4.

The package contains no concluding chapter since it is supposed to be extended in the future.

1.1 The Intrinsic Dimensionality

In [6] signals are distinguished according to the rank of the corresponding orientation tensor and according to be (non-) simple. In a strict sense, this means to mix up two different concepts: the rank is a property of a particular estimation method, whereas being a simple signal is a matter of the signal modeling. Actually, the signal itself has something like a 'rank', independently of the particular estimation method. This leads us to the concept of the *intrinsic dimensionality* of a signal.

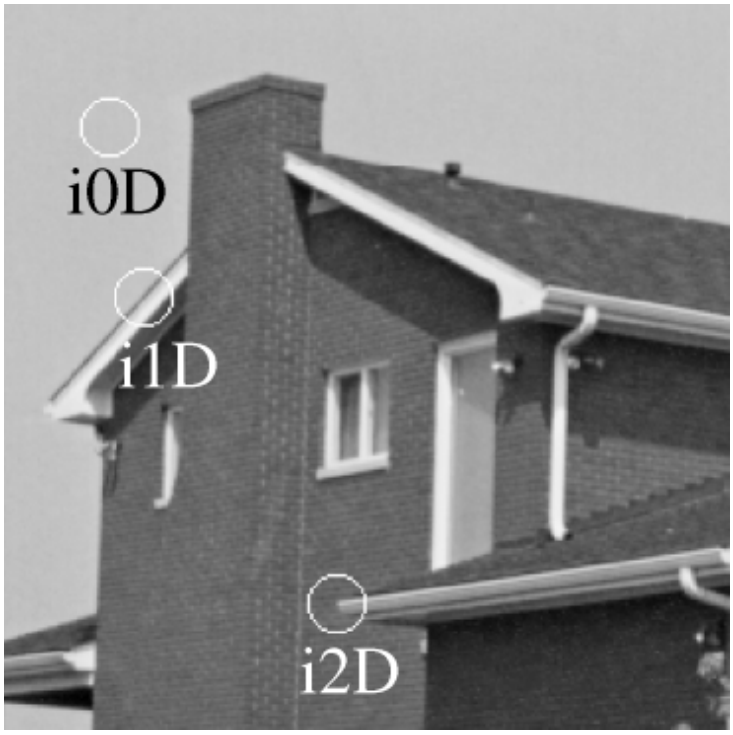
The latter is commonly defined as: "a data set in d dimensions is said to have an *intrinsic dimensionality* equal to d' if the data lies entirely within a d'-dimensional subspace" [1], p. 314.

In context of signal processing this leads us to the question of the considered domain: spatial (spatio-temporal) domain or Fourier domain? Since the Fourier transform replaces shifts with modulations, the Fourier domain is preferable. However, we want to consider local signal neighborhoods, and therefore we make use of the local Fourier transform (windowed Fourier transform).

Thus. we get the following categorization:
An nD signal has an intrinsic dimensionality of

- zero, if the local spectrum is concentrated in a point,
- one, if the local spectrum is concentrated in a line (this case is called simple signal in [6]),
- two, if the local spectrum is concentrated in a plane,
- ...,
- $n - 1$, if the local spectrum is concentrated in an $(n - 1)$D hyperplane,
- n, in all other cases.

For images, the different cases of intrinsic dimensionality can be illustrated, see figure 1.



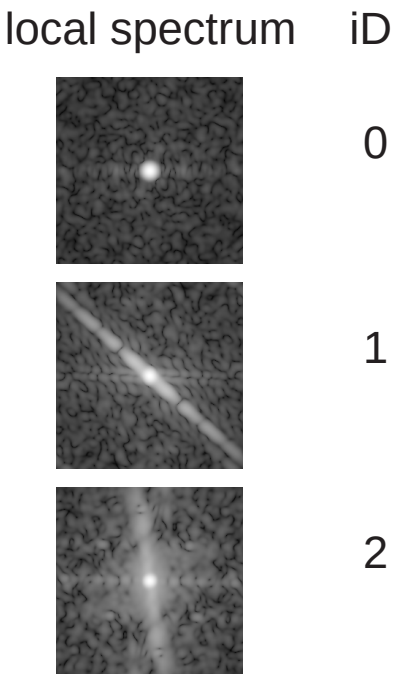


Figure 1: Illustration intrinsic dimensionality. In the image on the left, three neighborhoods with different intrinsic dimensionalities are indicated. The other three images show the local spectra of these neighborhoods.

Obviously, there is a one-to-one correspondence between rank k signals from [6] and signals with intrinsic dimensionality k. Anyhow, it is important to realize that the *rank of the orientation tensor* is an estimate of the signal rank, or intrinsic dimensionality, and not necessarily equivalent. Therefore, the term of intrinsic dimensionality is helpful to keep signal modeling and representation apart.

Furthermore, real signals are hardly *exactly* of a certain dimensionality. Instead, one should rather say that a signal is to a certain degree of a certain dimensionality, e.g., 95% of the signals energy is concentrated in a line. This degree of dimensionality is a signal property and not an effect of the estimation algorithm, caused by, e.g., noise, quantization, numerical error, etc. Opposed to that, the orientation tensor is commonly not a pure rank k tensor, which is in general caused by both, the signal property and the measuring process.

1.2 Stochastic Processes

There are two areas where stochastic processes are applied in signal processing. First, real signals are noisy caused by, e.g., sensor noise, motion blur, etc. When we estimate features of a signal we try to get rid of the noise and to extract the 'true' information (whatever that means). Second, certain signal properties can be modeled as stochastic processes, e.g., textures, orientation, etc.

Without going into too much mathematical details, a *stochastic process* P (for higher dimensional signals also called *random field*) can be considered as a stochastic generator of signals. At every spatial (spatio-temporal) position, the process corresponds to a *stochastic variable* and the observed signal value $f(\mathbf{x})$ corresponds to a particular realization of the stochastic variable ('a sample drawn from a set'). The whole observed signal is a particular realization of the stochastic process $P(\mathbf{x}, f)$.

In a sloppy formulation, a stochastic process is represented by the 'infinitesimal probability' (density) of observed function values $f(\mathbf{x}_0)$ at positions $\mathbf{x} = \mathbf{x}_0$. For 1D signals, this can be illustrated by drawing the density function, see figure 2. Considering the stochastic process at a particular position, this density is the (1D) *probability density function* (pdf) of the corresponding stochastic variable.

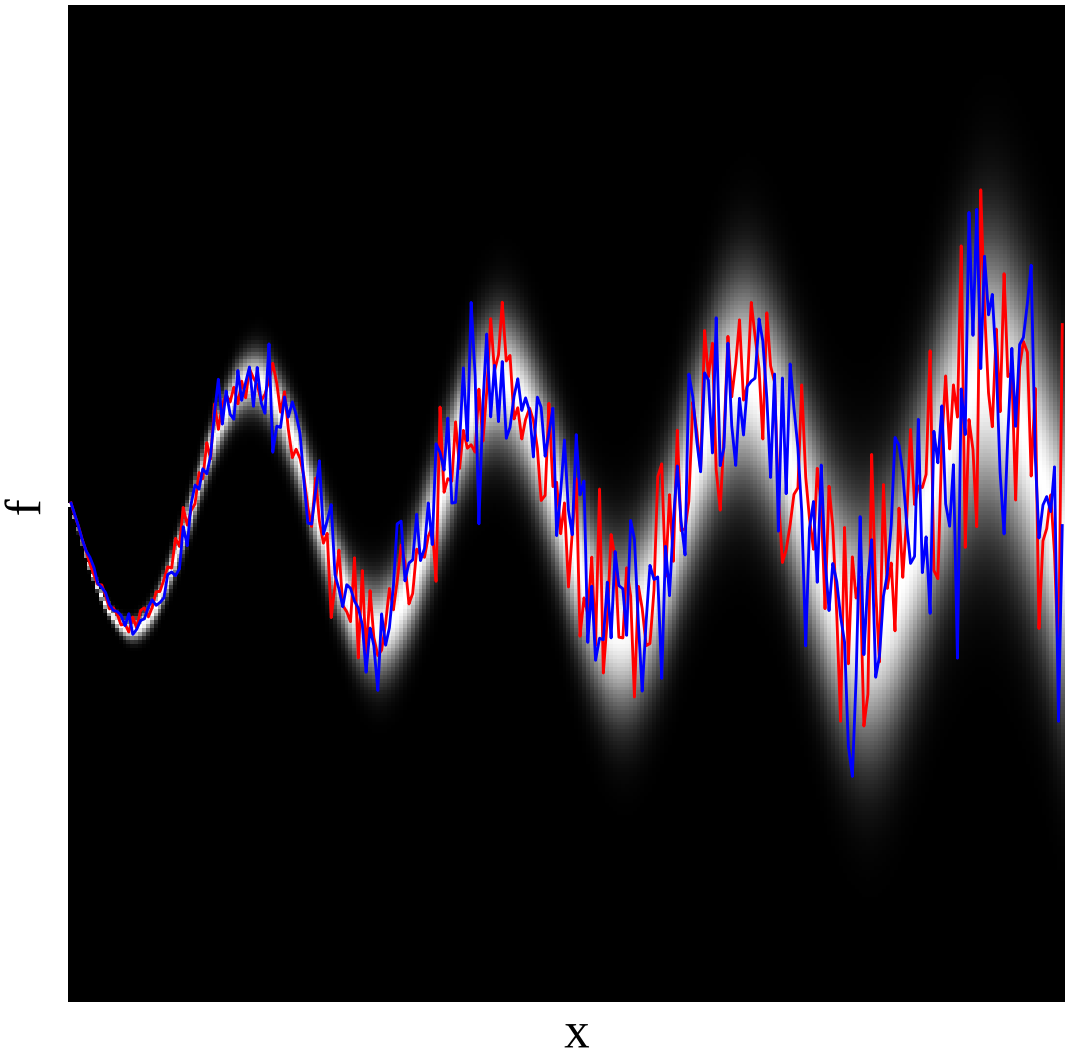


Figure 2: 1D stochastic process (density image) and two realizations (red and blue graph). The process is given by a sine curve with increasing variance, i.e., a sine curve with added Gaussian noise. The density is not normalized; white means one, black means zero.

If we knew all details about the stochastic process, we could compute the probability for signal values falling into a certain range by integrating the pdf over that range, we could compute the probability for the whole signal by integrating along the trajectory, or we could compute the expectation value at every position. The expectation value corresponds to the most likely signal value and having the expectation values at all positions means to know the realization which is most likely to be the 'real' signal.

However, without further assumptions, the only information about the process we have is a single realization, i.e., we know one sample for every pdf, which is insufficient for computing, e.g., the expectation values. This changes as soon as we introduce the assumption of *stationarity*. Stationarity of a stochastic process implies that the probability density is independent of **x**, i.e., the process is constant over **x**. These processes do not appear to be of much interest, but we will weaken the requirement to local regions further below.

Assuming that we have a stationary process, we can estimate arbitrary moments through integration of a single realization by applying the *ergodic hypothesis*. The latter assumes that the parametric mean value (integration wrt. **x**) is the same as the ensemble mean value (integration using the pdf) with a probability of one. In particular, we can compute the expectation value (which is independent of **x** since the process is stationary) by computing the mean value of the realization.[1]

As pointed out above, constant stochastic processes are not very interesting and hence, the stationarity assumption in its strict sense is not applicable to most signals. However, we can often assume that a process is stationary in a certain region, e.g., a neighborhood in an image is nearly constant or the local orientation is similar along an edge. We call this *local stationarity* in the following.

[1] If we are only interested in the expectation value, it is sufficient that the process is stationary in a weak sense [9].

For locally stationary processes we can assume the ergodic hypothesis to be valid in regions of stationarity. But what does that mean in practice?

Assume that our signal contains zero mean additive Gaussian noise. The expectation value of the stochastic process thus equals the noise-free signal. We can approximate this expectation value by averaging the signal values in the stationary regions. Unfortunately, these regions are unknown in advance, such that the region averaging is in practice replaced with a local Gaussian smoothing. At region boundaries, however, even local averaging makes no sense at all, since the process is instationary in the region of the filter support and we cannot assume ergodicity. From this perspective, linear averaging of a signal itself should be avoided in many cases, which is yet not true for averaging other signal representations.

2 Structure Tensor based on Gradients

In this chapter we will introduce a different way to compute the tensor representation of the local signal orientation. It is noteworthy that the tensor itself can be computed by various algorithms and that the quadrature filter based approach in [6] is not the only way to obtain the orientation tensor. In particular, we address the tensor computed from outer products of gradients, often referred to as *structure tensor*.

2.1 Approximating the Autocovariance Function

The structure tensor can be derived from locally approximating the autocovariance function of a real valued stochastic signal $f(\mathbf{x})$ (f is generated by a stochastic process) in the origin. This autocovariance function is defined by

$$C(\mathbf{x}_1,\mathbf{x}_2) = R(\mathbf{x}_1,\mathbf{x}_2) - m(\mathbf{x}_1)m(\mathbf{x}_2) \quad , \tag{1}$$

where

$$m(\mathbf{x}) = E[f(\mathbf{x})] \tag{2}$$

is the mean value of $f(\mathbf{x})$ and

$$R(\mathbf{x}_1,\mathbf{x}_2) = E[f(\mathbf{x}_1)f(\mathbf{x}_2)] \tag{3}$$

is the autocorrelation of $f(\mathbf{x})$.

Assuming stationarity, we obtain shift invariance of the autocovariance function, i.e., $C(\mathbf{x}_1+\mathbf{x}_0,\mathbf{x}_2+\mathbf{x}_0) = C(\mathbf{x}_1,\mathbf{x}_2)$ for all $\mathbf{x}_0$ and in particular $\mathbf{x}_0 = -\mathbf{x}_2$. Therefore, we consider $C(\mathbf{x})$ with $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_2$ in the following. The autocovariance function $C(\mathbf{x})$ can be estimated by local averages due to the ergodicity hypothesis.

Assuming ergodicity in the local region Ω, $m(\mathbf{x}) = m_\Omega$ (constant over $\mathbf{x}$), $C(\mathbf{x})$, and $R(\mathbf{x})$ can be computed by averaging over Ω. In particular, we obtain

$$m_\Omega = \frac{1}{|\Omega|} \sum_{\mathbf{x} \in \Omega} f(\mathbf{x}) \tag{4}$$

$$\begin{aligned} C(\mathbf{x}) &= \frac{1}{|\Omega| - 1} \sum_{\mathbf{x}' \in \Omega} (f(\mathbf{x}') - m_\Omega)(f(\mathbf{x}' + \mathbf{x}) - m_\Omega) \\ &= \frac{1}{|\Omega| - 1} f(\mathbf{x}) * f(-\mathbf{x}) - \frac{|\Omega|}{|\Omega| - 1} m_\Omega^2 \end{aligned} \tag{5}$$

$$R(\mathbf{x}) = \frac{1}{|\Omega| - 1} f(\mathbf{x}) * f(-\mathbf{x}) \tag{6}$$

where $*$ denotes convolution. The normalization by $|\Omega| - 1$ instead of $|\Omega|$ is necessary in order to obtain an unbiased estimate [1].

Even though the convolution formulation for $C(\mathbf{x})$ reduces the computational complexity significantly, it is still much too high for practical purposes, since the filter mask is potentially infinitely large.

Therefore, we approximate the autocovariance function by a local series expansion:

$$C(\mathbf{x}) = C(0) + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + O(\mathbf{x}^4) \tag{7}$$

where $\mathbf{H}$ is the Hessian of $C(\mathbf{x})$ in the origin, i.e.,

$$\mathbf{H} = \left[\frac{\partial^2}{\partial x_i \partial x_j} C(\mathbf{x})\right]_{ij}\Bigg|_{\mathbf{x}=0} \quad . \tag{8}$$

Apparently, the Hessian contains all orientation dependent terms of the series approximation of $C(\mathbf{x})$.

The fundamental equivalence for the calculation of the Hessian is the following:

$$H_{ij} = C_{ij}(0) \quad , \tag{9}$$

where $C_{ij}(\mathbf{x})$ is the *covariance function* of the partial derivatives of $f(\mathbf{x})$. The latter is obtained as

$$C_{ij}(\mathbf{x}) = \frac{1}{|\Omega| - 1} f_i(\mathbf{x}) * f_j(-\mathbf{x}) \tag{10}$$

where $f_i(\mathbf{x})$ are the partial derivatives $\frac{\partial f(\mathbf{x})}{\partial x_i}$. In order to show that H_{ij} and $C_{ij}(0)$ are indeed equivalent, we switch to the Fourier domain.

The nD Fourier transform of the autocorrelation function yields the *power spectrum* of $f(\mathbf{x})$

$$\mathcal{F}[R(\mathbf{x})] = \mathcal{F}[f(\mathbf{x}) * f(-\mathbf{x})] = F(\mathbf{u})F^*(\mathbf{u}) = P(\mathbf{u}) \ , \qquad (11)$$

where $F(\mathbf{u}) = \mathcal{F}[f(\mathbf{x})]$ is the nD Fourier transform of $f(\mathbf{x})$.

Hence, the Fourier transform of the Hessian is obtained as

$$\mathcal{F}[H_{ij}] = \mathcal{F}\left[\left.\frac{\partial^2 C(\mathbf{x})}{\partial x_i \partial x_j}\right|_{\mathbf{x}=0}\right] = \int -u_i u_j F(\mathbf{u})F^*(\mathbf{u})\, d\mathbf{u} \qquad (12)$$

and due to the power theorem it follows

$$-\int (iu_i F(\mathbf{u}))(-iu_j F^*(\mathbf{u}))\, d\mathbf{u} = -\int f_i(\mathbf{x}) f_j(\mathbf{x})\, d\mathbf{x} \ . \qquad (13)$$

That means, the part of Equation (7) that contains all orientation related information is computed by integrating the outer product of the signal gradient:

$$\mathbf{H} = -\int (\nabla f(\mathbf{x}))(\nabla f(\mathbf{x}))^T\, d\mathbf{x} \ . \qquad (14)$$

2.2 The Structure Tensor

If we replace the integration in Equation (14) with a local averaging, we obtain a *local, matrix-based orientation representation*:

$$\mathbf{J}(\mathbf{x}) = g(\mathbf{x}) * (\nabla f(\mathbf{x}) \nabla^T f(\mathbf{x})) \quad , \tag{15}$$

the *structure tensor*. Before we focus on properties of the latter and introduce some features, we want to mention another way for deriving the same representation.

Stating the optimization problem

$$\int g(\mathbf{x}_0 - \mathbf{x}) \langle \nabla f(\mathbf{x}) \mid \hat{\mathbf{n}}(\mathbf{x}_0) \rangle^2 \, d\mathbf{x} \;\rightarrow\; \max \quad , \tag{16}$$

i.e., at every point $\mathbf{x}_0$ we are looking for a unit vector $\hat{\mathbf{n}}$ which best represents the local gradient orientation. Solving this maximization directly leads to Equation (15).

This result is not surprising, since the approximation in Equation (7) is of quadratic order. Hence, we consider the variances in different orientations.

On the other hand, Equation (16) minimizes the (weighted) sum of the squared sine of the orientation error:

$$\int g(\mathbf{x}_0 - \mathbf{x}) \|\nabla f(\mathbf{x})\|^2 \sin(\measuredangle\alpha)^2 \, d\mathbf{x} \;\rightarrow\; \min \;. \qquad (17)$$

From a geometric point of view, this minimization tries to find the 1D subspace with the minimum cumulated distance to the samples, see figure 3.

Thus, it appears reasonable to switch the coordinate system accordingly, i.e., to consider the solution in its *eigensystem*: We calculate the eigenvectors and eigenvalues. This procedure is equivalent to the one described in [6], so we skip it here.

Another interesting aspect are the measures of the degree of simplicity, the *coherence*, of the signal. We would like to express something like the *curvature of the autocovariance function* in terms of coefficients.

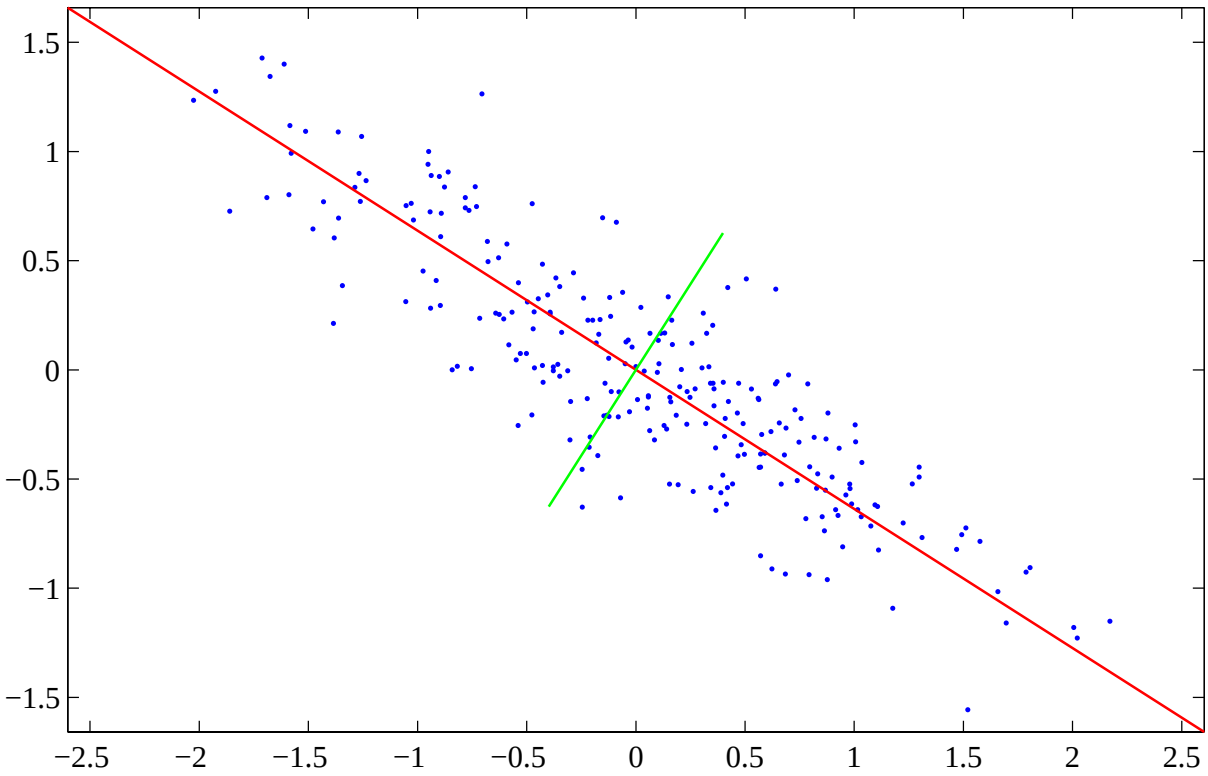


Figure 3: Cluster of points (blue) and corresponding 1D subspace with minimum distance (red) and maximum distance (green).

In the 2D case, this is comparably simple: we consider the *Gaussian curvature* C_g, which describes the type of curvature (elliptic, paraboloid, planar), and the *mean curvature* C_m. These measures are related to the eigenvalues λ_1 and λ_2 of the Hessian by

$$\lambda_{1/2} = C_m \pm \sqrt{C_m^2 - C_g} \ , \tag{18}$$

i.e., the eigenvalues are increasingly different of increasing ratios

$$\frac{\sqrt{C_m^2 - C_g}}{C_m} = \frac{\sqrt{(\lambda_1 + \lambda_2)^2 - 4\lambda_1\lambda_2}}{\lambda_1 + \lambda_2} = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \ . \tag{19}$$

This expression describes the coherence of the structure and is an alternative to (51) in [6].

3 Isotropic Quadrature Filters

work in progress

4 Multiple Orientation Representation by Channels

to be added later

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.

[2] M. Felsberg. *Low-Level Image Processing with the Structure Multivector*. PhD thesis, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel, 2002. TR no. 0203, available at http://www.informatik.uni-kiel.de/reports/2002/0203.htm

[3] W. Förstner. *Statistische Verfahren fur die automatische Bildanalyse und ihre Bewertung bei der Objekterkennung und -vermessung*. Number 370 in C. Verlag der Bayerischen Akademie der Wissenschaften, 1991.

[4] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, Dordrecht, 1995.

[5] B. Jähne. *Digital Image Processing*. Springer, Berlin, 2002.

[6] H. Knutsson. *Edupack: Orientation*. LiU, April 2000. available at http://www.isy.liu.se/cvl/Education/Edupack/index.html.

[7] A. Papoulis. *The Fourier Integral and its Applications*. McGraw-Hill, New York, 1962.

[8] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1965.

[9] A. Papoulis. *Signal Analysis*. McGraw-Hill, 1977.